Visão

Com a crescente demanda sobre Tecnologias, percebemos que muitas pessoas apesar de buscarem informações, não possuem fontes que queiram realmente passar o conhecimento da maneira como ela deve ser, livre e com embasamento técnico que permita ser aplicado e utilizado quando necessário, além de serem testados em sua criação, tornando esta informação útil e confiável.

Missão

O Laboratório foi criado com a intenção de buscar e disseminar o conhecimento de uma maneira clara e objetiva, de forma gratuita, auxiliando na evolução dos membros e da sociedade na qual estas informações são compartilhadas, buscando o crescimento de todos os envolvidos nesta criação de valores.

Licença

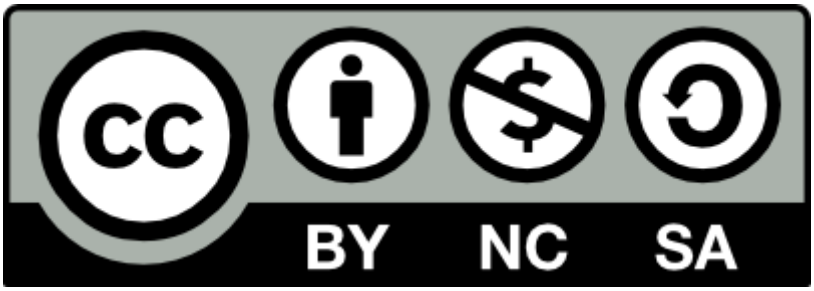

Figura 01 – Licença Criative Commons – by-nc-as

Esta licença permite que outros remixem, adapte, e criem obras derivadas sobre a obra original, desde que com fins não comerciais e contanto que atribuam crédito ao autor e licenciem as novas criações sob os mesmos parâmetros. Outros podem fazer download ou redistribuir a obra da mesma forma que na licença anterior, mas eles também podem traduzir, fazer remixes e elaborar novas histórias com base na obra original. Toda nova obra feita a partir desta deverá ser licenciada com a mesma licença, de modo que qualquer obra derivada, por natureza, não poderá ser usada para fins comerciais.

This license lets other remix, tweak, and build upon your work non-commercially, as long as they credit you and license their new creations under the identical terms.

Para maiores informações sobre o método de licenciamento acesse os seguintes sites:

Brasil:
http://creativecommons.org.br/as-licencas/
http://creativecommons.org/licenses/by-nc-sa/3.0/br/

Internacional:
http://creativecommons.org/licenses/by-nc-sa/3.0/
http://creativecommons.org/licenses/by-nc-sa/3.0/legalcode

1 – PAM (Pluggable Authentication Modules)

O **PAM** é um mecanismo de autenticação que teve como origem no **Unix Solaris**, mas que é utilizado em vários sistemas, inclusive no **Linux**. Ele torna o processo de autenticação mais flexível, dando suporte a diversas aplicações com ou sem suporte ao sistema de **login** oferecido no **Linux**. São através dele que atribuímos políticas de contas, logins, sessões e senhas.

2 – Remover Arquivo de Compatibilidade

Antes mesmo de explicarmos como utilizar os módulos do **PAM,** vamos retirar alguns arquivos desnecessários, para deixarmos nosso sistema mais limpo e também evitarmos problemas causados por arquivos de compatibilidade.

Originalmente, o **PAM** usava o ***/etc/pam.conf*** como seu arquivo de configuração. Mas, hoje em dia, esse arquivo só é necessário se o diretório de configuração padrão do **PAM** não existir. Portanto devemos remover o arquivo ***pam.conf***, já que todas as informações de configuração do **PAM** estão em um único diretório, denominado ***/etc/pam.d***.

Cada serviço tem seu próprio arquivo de configuração no ***/etc/pam.d***, e o nome desse arquivo corresponde ao do programa ou serviço. Por exemplo, o aplicativo de **logon (/bin/login)** está configurado em /**etc/pam.d/login**. Os programadores definem os nomes dos serviços ou aplicativos.

```
root@fusion:~# cat /etc/pam.conf

# ----------------------------------------------------------------------#
# /etc/pam.conf                                                         #
# ----------------------------------------------------------------------#
#
# NOTE
# ----
#
# NOTE: Most program use a file under the /etc/pam.d/ directory to setup
their
# PAM service modules. This file is used only if that directory does not
exist.
# ----------------------------------------------------------------------#

# Format:
# serv. module     ctrl       module [path]     ...[args..]          #
# name  type       flag                                              #

root@fusion:~# rm /etc/pam.conf
root@fusion:~#
```

3 – Sintaxe dos Arquivos de Configuração do PAM

Cada linha do arquivo de configuração do **PAM** é formada ou por uma linha de comentário que inicia com um **#** ou por uma linha com quatro possíveis argumentos: uma **interface de módulo**, um **sinalizador de controle**, um **caminho do módulo** e **argumentos do módulo**.

Ficando assim:

```
interface          controle     caminho_mod        [argumentos_mod]
```

Observação: Os argumentos do módulo são opcionais e dependem de cada modulo.

4 – Interface do Módulo

Na terminologia do **PAM**, a expressão "**interface do módulo**", corresponde ao tipo de autorização de um módulo. As interfaces são: **account**, **auth**, **password** e **session**. Cada uma delas será especificada no arquivo de configuração do serviço, caso ele tenha suporte.

- **account** → A interface **account** verifica se uma conta tem autorização para usar o sistema, o que pode significar, verificar se um serviço existe, se a conta está bloqueada ou expirada, se o acesso tem restrições quanto a horários de acesso, etc.
- **auth** → A interface **auth** autentica um usuário. Isso pode ser feito solicitando uma senha, um banco de dados ou outro mecanismo e, em seguida, verificando-o. Os módulos **auth** também têm permissão para definir credenciais, como membros de grupo ou tíquetes **Kerberos**.
- **password** → A interface **password** é usada para verificar e definir a autenticação de senha.
- **session** → A interface **session** configura e gerencia uma sessão de usuário, pós autenticação. Isso pode incluir tarefas nas sessões, como montar diretórios, criar arquivos, etc.

5 – Sinalizador de Controle

Para cada interface, o arquivo de configuração especifica um sinalizador de controle, que determina o que o **PAM** fará em seguida, com base no resultado da verificação realizada. Existem quatro sinalizadores de controle: **optional**, **required**, **requisite** e **sufficient**.

- **optional** → Os módulos opcionais não afetam o sucesso nem a falha da autenticação, a menos que não haja outros módulos para determinada interface.
- **required** → Para que o usuário possa continuar, deverá ser retornado um resultado de sucesso. A notificação de usuário não ocorre até que todos os módulos para a interface estejam satisfeitos ou verificados.
- **requisite** → Para que o usuário possa continuar, deverá ser retornado um resultado de sucesso. A notificação de usuário acontece imediatamente em caso de falha no primeiro módulo **requisite** ou **required** de uma interface.
- **sufficient** → Um resultado de sucesso, combinado a nenhuma falha do módulo **required,** viabiliza uma boa autenticação. A falha de um módulo **sufficient** é ignorada.

6 – Caminho do Módulo

O caminho do módulo informa ao **PAM** a localização do módulo. Quando não informado, o **PAM** busca no caminho padrão dos módulos, que é ***/lib/security***, neste caso informamos apenas o nome do módulo.

7 – Argumentos do Módulo

Se um módulo exigir argumentos, é aí que eles entrarão. Serão algo como **file=/etc/ftp.allow**. Os argumentos inválidos não prejudicam o processo de logon e são facilmente registrados no ***syslog***.

Observação: Antes de fazer qualquer alteração em um arquivo de configuração do **PAM**, faça o backup antes, garantindo que se houver algo errado nas configurações efetuadas, o retorno para as regras anteriores seja imediato.

8 – Restaurando o PAM Após Um Erro Grave

Se por acidente, o seu sistema não inicializa mais devido a algum erro no **PAM,** podemos fazer o seguinte procedimento para retornar rapidamente e depois analisarmos os arquivos de logs atrás do problema.

```
root@fusion:~# rm -r /etc/pam.d/

root@fusion:~# reboot

fusion login: root

Login incorrect

Login incorrect

Login incorrect

Login incorrect

Login incorrect

fusion login:
```

Observe que ele informa erros de login pela ausência do arquivo ***/etc/pam.d/login***. Caso não tenha backup dos arquivos, não entre em pânico, basta reiniciar o sistema em modo **sigle** e criar o diretório ***/etc/pam.d*** e dentro dele o arquivo ***login*** da seguinte forma:

```
root@fusion:~# mkdir -p /etc/pam.d
root@fusion:~# cat > /etc/pam.d/login
auth            required        pam_unix.so
account         required        pam_unix.so
password        required        pam_unix.so
session         required        pam_unix.so
^C
root@fusion:~# init 2

--==[ Resumido ]==--

fusion login: root
password:

root@fusion:~#
```

Esta é a configuração mais básica do **PAM**. Após isso, devemos reinstalar o **PAM** para que ele possa vir com as configurações adequadas para nossa distribuição.

9 – Referencias Bibliográficas

[1] Morimoto, Carlos. Disponível em: <http://www.hardware.com.br/termos/bios>. Acessado em: 21/07/2012.

[2] Duarte, Helto, <http://heltonduarte.com/2009/07/03/gerenciadores-de-boot/> Acessado em 21/07/2012.

[3] Ribeiro, Uira – Certificação Linux, 1º Ed, São Paulo, 2004, Axcel Books

[4] Manual do GNU GRUB v2. Disponível em: <http://www.gnu.org/software/grub/manual/html_node/Security.html#Security>. Acessado em: 24/07/2012.

[5] Drs305. Disponível em: <http://ubuntuforums.org/showthread.php?t=1369019>. Acessado em: 24/07/2012.

[6] Artigo. Disponível em: <http://www.linuxhowtos.org/Tips%20and%20Tricks/sysrq.htm> . Acessado em : 28/07/2012.

[7] Manual Debian. Disponível em: <http://www.debian.org/doc/manuals/debian-reference/ch09.pt.html#_alt_sysrq_key>. Acessado em: 31/07/2012.

[8] Documentação Kernel. Disponível em: <http://Kernel.org/doc/Documentation/sysrq.txt>. Acessado em: 31/07/2012.

[9] Debian Administrator. Disponível em: <http://www.debian-administration.org/article/457/The_magic_sysreq_options_introduced> . Acessado em: 28/07/2012.

[10] Terpstra, John; Love, Paul; Reck, Ronald; Scanlon, Tim – Segurança para Linux, 1º Ed, 2005,